A parametric flow-based algorithm for crew diagramming: a pragmatic large-scale approach at the Swiss railways

Raphael Haemmerli * Alexander Souza (Speaker) * Denys Trieskunov *
Hsuan-Pin Wu *

1 Introduction

One of the central planning challenges in the realm of mass transportation, is to determine shifts for crews (located at a depot), e.g., drivers, pilots, or conductors. A shift, also called a *crew diagram*, describes the activities, called *crew requirements*, the crew has to execute during a day of operations, e.g., setup-, driving-, shutdown-, passride-, and break-tasks. Labor rules must be respected to ensure safe transportation services.

Mathematically, the Crew Diagramming problem asks to partition a set of crew requirements into a set of crew diagrams, while respecting constraints, such as chronological consistency, local consistency (bridging gaps by passrides), maximum duration of uninterrupted work (ensured by breaks), maximum diagram duration, and start- and end-location at the depot of the crew. The objective is to minimize a weighted sum of the number of crew diagrams and further problem data. It is related to Set Cover, respectively Set Partition, see [1], and is NP-complete (in its decision version).

Crew Diagramming has attracted substantial research from the operations research community, see, e.g., [3, 8, 10, 11, 5, 6, 4, 9, 7]. The dominant approach is column-generation [12], with Set Cover as the master problem [3, 10, 5, 4, 9] and Resource Constrained Shortest Path (RCSP) as the subproblem [14, 5, 13, 9].

An operational optimizer, which is based on column-generation, was rejected by the planning department at the Swiss railways, because of several reasons: the runtimes have been unsatisfactory, especially for medium and large depots; the quality of the solution deteriorated because of necessary postprocessing converting multiply covered crew requirements into passrides; and the variability of the solutions due to built-in randomization.

2 A parametric flow-based algorithm

This research originates from our work at the Swiss railways, where we designed and implemented a new optimizer for Crew Diagramming. Our goal has been to address the reasons leading to the rejection of the existing optimizer. In our approach, we introduce a variety of parameters for our algorithm, allowing the planners to balance between runtime, feasibility, and optimality. Hence we leave it up to the users, which

* { raphael | alexander | denys | hsuan-pin }@algomia.com Algomia GmbH, Schachenallee 29, CH-5000 Aarau, Switzerland.

aspect of the optimization they want to emphasize. In our experience, speed is by far the most important feature, followed by feasibility and finally optimality. The main reason is that planners always adjust the results of the optimizer, e.g., to incorporate soft-knowledge or preferences that are not reflected therein.

Even during the development, we tested our prototypes on real-world data. This allowed us to reject approaches that performed poorly already early on. Furthermore, our solutions have been constantly validated by super-users, leading to incorporation of soft-knowledge and a very good acceptance by the customer.

The basic idea of the algorithm is to combine a state-expanded Minimum Cost Flow formulation with additional Set Partition constraints. A source vertex in the flow network represents the start of the planning day at the depot, while a target represents the end there. Any other vertex represents the execution of a crew requirement in a crew diagram and contains additional values, called resources. These represent (the relevant part of) the history of the vertex, e.g., the start of its crew diagram, the end of the last break in the diagram, and so on. If there exists an edge uv, between vertices u and v, then the crew requirement of v can be executed after u, and the resource-values are propagated accordingly. The edges carry supporting information, like, breaks or passrides that are planned between crew requirements. By construction, each path from the source to the target represents a valid crew diagram. Capacity constraints ensure that at most one unit of flow passes through each vertex. Hence the entire flow decomposes into disjoint paths. Set Partition constraints ensure that each crew requirement occurs in exactly one vertex with positive flow.

The algorithm has several parameters that are aimed to balance between optimality, feasibility, and speed. Clearly, the state-expansion has to be done with care, because it can lead to combinatorial explosion of the network, otherwise. We do so by introducing discretization parameters that lead to “rounding” of the resource-values, thus significantly reducing the state-space. Of course, we only generate states that can be reached from the source. Furthermore, we find that the LP-relaxation of our formulation is very strong empirically: usually, the LP- and ILP-optima are only a few percent away from each other. To accelerate the algorithm even further, we shrink the ILP after the LP-relaxation with vertex-elimination heuristics. In case of multi-depot/multi-group planning, the algorithm allows the user to compute a pre-assignment of the crew requirements, which leads to a further significant speed-up (at the potential expense of optimality). It is furthermore possible to split large instances by solving a relaxed version of the problem. All these parametrization options lead to attractive runtimes in practice with good solution qualities.

The following results have been obtained with a cloud-instance having 128 GB of main memory and 28 CPU cores.

Depot (Groups)	CRs	Manual CDs	Optimizer CDs	Δ	Runtime [s]
BEL (1)	111	11	11	0	12
BI (1,2)	699	47	44	-3	330
BS (1,2)	372	30	30	0	845
CH (1)	208	18	18	0	10
LS (1,2,3)	579	40	39	-1	135
LZ (1,2,3,4)	688	56	57	+1	968
OL (1,2)	817	58	55	-3	2412
ZUE (1,2,3,4)	1016	80	80	0	5264

In conclusion, our algorithm is capable of solving all 36 depots at the Swiss railways sequentially in about 3 hours. Also simultaneous multi-depot optimization is possible. All of the criticsm against the existing optimizer has been addressed: our algorithm has good runtimes, does not need postprocessing, and is deterministic. The most important feature is that it works well in practice.

References

[1] M.R. Garey and D.S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco.

[2] R. Kwan (2011). Case studies of successful train crew scheduling optimisation. J. Scheduling. 14(5). 423-434. pp. 49-54, 2011.

[3] B.M. Smith and A. Wren (1988). *A bus crew scheduling system using a set covering formulation.* In: Transportation Research, vol. 22A, pp. 97-108, 1988.

[4] S. Fores and L. Proll and A. Wren (2002). *TRACS : a hybrid IP/heuristic driver scheduling system for public transport.* Journal of the Operational Research Society, 53:1093-1100, 2002

[5] A.S.K. Kwan (1999). *Train Driver Scheduling*, PhD Thesis, University of Leeds

[6] G. P. Silva and N. D. F. Gualda and R. Kwan (2000). *Bus Scheduling Based on an Arc Generation – Network Flow Approach*, 8th International Conference on Computer-Aided Scheduling of Public Transport - CASPT 2000

[7] A. Kwan and M. Parker and R. Kwan and S. Fores and A. Wren (2022). *Recent advances in TRACS*, Technical Report, University of Leeds, 2022.

[8] J.E. Beasley and B. Cao (1996). *A tree search algorithm for the crew cheduling problem*, J. Operational Research 94, pp. 517 – 526, 1996.

[9] S. Jütte (2012). *Large-Scale Crew Scheduling: Models, Methods, and Applications in the Railway Industry*, Springer Verlag, 2012.

[10] R. Borndörfer, M. Grötschel, and A. Löbel (1998). *Optimization of Transportation Systems.* Technical Report ZIB-Report 98-09, Zuse-Institut Berlin, 1998

[11] R. Borndörfer, A. Löbel, U. Strubbe, and M. Völker (1999). *Zielorientierte Dienstplanoptimierung.* In: Heureka '99: Optimierung in Verkehr und Transport, pp. 171194, Köln, 1999. ZIB Report 98-41.

[12] G. B. Dantzig and P. Wolfe (1960). *Decomposition Principle for Linear Programs.* Operations Research, 8(1):101111, 1960.

[13] S. Irnich and G. Desaulniers (2005). *Shortest Path Problems with Resource Constraints.* In: Desaulniers G., Desrosiers J., Solomon M.M. (eds) Column Generation. Springer, Boston, MA, 2005.

[14] M. Desrochers (1986). *La Fabrication dhoraires de travail pour les conducteurs dautobus par une mthode de gnration de colonnes.* Ph.D Thesis, Centre de recherche sur les Transports, Publication 470, Universit de Montral, Canada, 1986.